

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2004-178015

(P2004-178015A)

(43) 公開日 平成16年6月24日(2004.6.24)

(51) Int.Cl.⁷
G06F 12/00F 1
G06F 12/00 547A
G06F 12/00 515Bテーマコード(参考)
5B082

審査請求 未請求 請求項の数 15 O L (全 16 頁)

(21) 出願番号
(22) 出願日特願2002-340041 (P2002-340041)
平成14年11月22日 (2002.11.22)(71) 出願人 000003078
株式会社東芝
東京都港区芝浦一丁目1番1号
(74) 代理人 100058479
弁理士 鈴江 武彦
(74) 代理人 100084618
弁理士 村松 貞男
(74) 代理人 100068814
弁理士 坪井 淳
(74) 代理人 100092196
弁理士 橋本 良郎
(74) 代理人 100091351
弁理士 河野 哲
(74) 代理人 100088683
弁理士 中村 誠

最終頁に続く

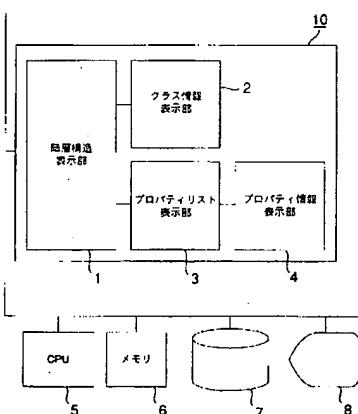
(54) 【発明の名称】階層構造表示装置および階層構造表示方法

(57) 【要約】

【課題】階層型データベースにおけるクラス間の包含関係を適切かつ効果的に表示することができる階層構造表示装置および方法を提供すること

【解決手段】階層構造をなすクラスが定義された階層型データベースの階層構造表示を表示するにあたり、階層構造をなすいづれか一のクラスを識別する第1の表示領域の少なくとも一部が、該クラスから派生した子クラスを識別する第2の表示領域の全てを含むように前記第1及び第2の表示領域を表示する。

【選択図】 図1



【特許請求の範囲】**【請求項 1】**

階層構造をなすクラスが定義された階層型データベースの階層構造表示装置において、前記階層構造をなすいずれか一のクラスを識別する第1の表示領域の少なくとも一部が、該クラスから派生した子クラスを識別する第2の表示領域の全てを含むように前記第1及び第2の表示領域を表示する階層構造表示手段を具備することを特徴とする階層構造表示装置。

【請求項 2】

選択された前記一のクラス及び子クラスのいずれかに関連するクラス情報を表示するクラス情報表示手段を具備することを特徴とする請求項1に記載の階層構造表示装置。

10

【請求項 3】

選択された前記一のクラス及び子クラスのいずれかが有するプロパティのリストを前記クラス情報と関連付けて表示するプロパティリスト表示手段を具備することを特徴とする請求項2に記載の階層構造表示装置。

【請求項 4】

前記プロパティリストから選択されたプロパティに関連するプロパティ情報を表示するプロパティ情報表示手段を具備することを特徴とする請求項3に記載の階層構造表示装置。

【請求項 5】

継承元のクラスと、継承先のクラスにおける前記プロパティリスト中のプロパティとを同一色で対応付けて表示する手段を具備することを特徴とする請求項3に記載の階層構造表示装置。

20

【請求項 6】

前記子クラスを識別する第2の表示領域の表示有無を前記一のクラスに対して与えられた開閉操作に応じて切り替える手段を具備することを特徴とする請求項1乃至5のいずれかに記載の階層構造表示装置。

【請求項 7】

インスタンスを有するクラスのみを展開表示する手段を具備することを特徴とする請求項1乃至6のいずれかに記載の階層構造表示装置。

【請求項 8】

前記階層構造内において選択された一のクラスの直系クラスのみを表示する手段を具備することを特徴とする請求項1乃至7のいずれかに記載の階層構造表示装置。

30

【請求項 9】

前記いずれか一のクラスに関連付けて、前記子クラスのいずれか一つがインスタンスを有するか否かを識別するマークを表示する手段を具備することを特徴とする請求項1乃至8のいずれかに記載の階層構造表示装置。

【請求項 10】

前記階層構造内のいずれか一のクラスを検索開始ポイントに設定する手段と、前記検索ポイントを設定するクラスをインスタンスを有するクラスに限定する手段とを具備することを特徴とする請求項1乃至9のいずれかに記載の階層構造表示装置。

40

【請求項 11】

前記検索ポイントを設定するクラスを、インスタンスを有する少なくとも2つ以上の子クラスを有するクラスに限定することを特徴とする請求項10に記載の階層構造表示装置。

【請求項 12】

前記階層型データベースは、複数の親クラスからのプロパティの多重継承を許容することを特徴とする請求項1乃至11のいずれかに記載の階層構造表示装置。

【請求項 13】

前記多重継承に係るプロパティを選択し、該プロパティの継承元に相当する前記親クラスのいずれかを参照する手段を具備することを特徴とする請求項12に記載の階層構造表示装置。

【請求項 14】

50

第1のルートを有する第1の階層構造と、第2のルートを有する第2の階層構造とに共通するユニバーサル・ルートを設定する手段を具備することを特徴とする請求項1乃至13のいずれかに記載の階層構造表示装置。

【請求項15】

階層構造をなすクラスが定義された階層型データベースの階層構造表示方法において、前記階層構造をなすいずれかのクラスを識別する第1の表示領域の少なくとも一部が、該クラスから派生した子クラスを識別する第2の表示領域の全てを含むように前記第1及び第2の表示領域を表示する階層構造表示ステップと、

前記一のクラス及び子クラスのいずれかの選択を受け付けるステップと、

前記選択に連動し、該選択された前記一のクラス及び子クラスのいずれかに関連するクラス情報を表示するクラス情報表示ステップと、

前記選択に連動し、該選択された前記一のクラス及び子クラスのいずれかが有するプロパティのリストを前記クラス情報と関連付けて表示するプロパティリスト表示ステップとを具備することを特徴とする階層構造表示方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、階層型データベースに好適な階層構造表示装置および方法に関する。

【0002】

【従来の技術】

今日、コンピュータソフトウェアの技術分野は、よりユーザフレンドリな機能を提供すべく日々進歩を遂げている。例えば、ファイルシステムの利用にあたりユーザが所望のディレクトリに移動して所望のファイルを開くような操作（オペレーション）を視覚的かつ直感的に行えるように、グラフィック・ユーザ・インターフェース（G U I）は線やアイコン等を組み合わせてディレクトリ（フォルダ）構造をツリー状に表示し、ユーザに提示することができる。このようなG U I機能はマイクロソフト社のオペレーティング・システム（O S）W i n d o w s（商標）や、その他にもU N I X（商標）やL I N U X（商標）などの汎用のオペレーティングシステムにおいて広く採用されている。

【0003】

ディレクトリ構造ツリーの各ノード（ディレクトリ）間について、上位のノードに含まれる情報（ファイル等）と下位のノードに含まれる情報との間に、包含あるいは部分集合といった関係はない。すなわち、ディレクトリ構造ツリーを形成する、ルート・ノードから始まる各ノードは、ファイルなどの情報を納めるコンテナの相互接続関係を表しているにすぎない。

【0004】

一方、オブジェクト指向データベース（O O D B）やオブジェクト・リレーションナルデータベース（O R D B）を代表とするデータベースは、階層構造を有し、該階層構造において下位分類が上位分類の属性を継承する仕組みを有する。このようなデータベースでは、継承によって下位の分類では属性が累増するという特徴がある。下位分類が上位分類の属性を継承することは、「インヘリタンス」とも呼ばれ、このような技術は多くの文献に記載されている（例えば下記非特許文献1参照。）。

【0005】

なお、オブジェクト指向データベース（O O D B）関連の技術分野では、階層中の分類を「クラス」と呼ぶことが多い。本明細書では、「分類」と「クラス」とをほぼ同じ意味を有する用語として用いる。また、オブジェクト・リレーションナル型データベース（O R D B）においては、継承を許したテーブルがクラスに相当する。上下関係にあるテーブルにおいては、上位のテーブルから下位のテーブルへ属性が継承される。ここでの属性は、O R D Bにおいては上位テーブルを構成するコラムのヘッダー情報に相当し、これが下位テーブルへ継承される。

【0006】

10.

20

30

40

50

本明細書においては、オブジェクト指向データベース（O O D B）とオブジェクト・リレーションナルデータベース（O R D B）の両者を含めて「階層型データベース」と称する。また、各階層のクラスに属する同じ属性種を持つデータを「インスタンス」と呼び、その集合をデータの「ポピュレーション」と呼ぶ。

【0007】

このように、階層型データベースの階層構造は、クラス間の包含関係（継承関係）があることから、上述のディレクトリ構造ツリーとはその性質が明らかに異なるものである。

【0008】

階層型データベースを製品カタログデータベースに応用したサービスが既に幾つか提供されている（下記非特許文献2，3を参照。）。

10

【0009】

【非特許文献1】

Object-Oriented Concepts, Databases, and Applications, Edited by Won Kim, 1989, ACM Press

【0010】

【非特許文献2】

株式会社ポータル、インターネット<URL: http://www.portalcorp.net>

20

【0011】

【非特許文献3】

e-ingBizz.com、インターネット<URL: http://www.e-ingbizz.com//CATALOG2/servlet/CatalogSearch?Lang=j a>

【0012】

【発明が解決しようとする課題】

階層構造表示に関し、従来のディレクトリ構造ツリーにおける表示、例えばマイクロソフト社（商標）のエクスプローラでは、

- ・下位のフォルダ（ディレクトリ）を上位のフォルダへの開閉要求に応じて展開表示する
- ・段下げにより、そのフォルダよりも上位のフォルダに対して下位のフォルダであることを表す
- ・階層のレベル（深さ）ごとに段下げを揃えることにより、レベルが同じフォルダであることが容易に分かるようにする

といったことが行える。

30

【0013】

しかしながら、これらはあくまでディレクトリ（ノード）の接続関係を辿ることができるようにするものでしかない。

【0014】

したがって、このような従来の階層構造表示は、上位のクラスから下位のクラスへプロパティ（属性）が継承されるような階層型データベースには適さない。

40

【0015】

階層型データベースにおける階層構造表示には、個々のクラスが、その所有するプロパティによって特徴付けられる概念を表す場合の、クラス間の包含関係を表現できることが求められる。また、階層型データベースの階層構造がクラスによって構成される場合、該クラスのインスタンスの存在有無を表示できることが好ましい。

【0016】

本発明はかかる事情を考慮してなされたものであり、階層型データベースにおけるクラス間の包含関係を適切かつ効果的に表示することのできる階層構造表示装置および方法を提供することを目的とする。

【0017】

50

【課題を解決するための手段】

本発明に係る階層構造表示装置は、階層構造をなすクラスが定義された階層型データベースの階層構造表示装置において、前記階層構造をなすいずれか一のクラスを識別する第1の表示領域の少なくとも一部が、該クラスから派生した子クラスを識別する第2の表示領域の全てを含むように前記第1及び第2の表示領域を表示する階層構造表示手段を具備することを特徴とする。

【0018】

本発明に係る階層構造表示方法は、階層構造をなすクラスが定義された階層型データベースの階層構造表示方法において、前記階層構造をなすいずれか一のクラスを識別する第1の表示領域の少なくとも一部が、該クラスから派生した子クラスを識別する第2の表示領域の全てを含むように前記第1及び第2の表示領域を表示する階層構造表示ステップと、前記一のクラス及び子クラスのいずれかの選択を受け付けるステップと、前記選択に連動し、該選択された前記一のクラス及び子クラスのいずれかに関連するクラス情報を表示するクラス情報表示ステップと、前記選択に連動し、該選択された前記一のクラス及び子クラスのいずれかが有するプロパティのリストを前記クラス情報と関連付けて表示するプロパティリスト表示ステップとを具備することを特徴とする。

【0019】

【発明の実施の形態】

以下、図面を参照しながら本発明の実施形態を説明する。

【0020】

図1は本発明の一実施形態に係る階層構造表示システムの概略構成を示すブロック図である。本システムは、階層型データベースの階層構造表示に基づいて、ユーザが検索やデータベース管理等を、その規模や詳細度等に依らず適切かつ容易に行えるようにするものである。本発明に係る階層構造表示装置は、CPU5、メモリ6、二次記憶装置7、ディスプレイ8等の基本ハードウェアを備えた汎用のコンピュータを用い、同コンピュータ上で動作するアプリケーションプログラム10として実現することができる。この場合、同コンピュータにはアプリケーションプログラム10の実行環境を提供するとともに、ハードウェアを制御するためのオペレーティングシステムも導入されている。階層構造表示装置のアプリケーションプログラム10は、オペレーティングシステム上に構築され、ディスプレイ8や図示しないマウス、キーボード等と協働するグラフィカルユーザインターフェース(GUI)を構成する。階層型データベースは、例えば二次記憶装置7に導入されている。

【0021】

同図に示すように階層表示装置は、階層構造表示部1と、クラス情報表示部2と、プロパティリスト表示部3と、プロパティ情報表示部4とにより構成されている。

【0022】

ISO13584に規定のように、プロパティなどのクラス特性を下位クラスが上位クラスから継承するような階層型データベースにおいては、階層中の単なる上位、下位の関係ではなく、包含関係として階層構造が表現されたほうが、利用者にとってはクラス間の関係がわかりやすい。

【0023】

そこで、階層構造表示部1は、図2に示すようにクラス間の包含関係を明示的に表示する。具体的には、上位クラスが下位クラスを包含するように矩形等のダイアグラムを用いてクラスを表示する。

【0024】

図2の表示例によれば、"Product"クラス20が"Vehicle"クラス21を包含していることが明示される。この"Vehicle"クラス21は、"Automobile"クラス22および"Automobile"クラス23の2つのクラスを含んでいることがわかる。言い替えれば、"Automobile"クラス22および"Automobile"クラス23の2つのクラスは、"Vehicle"クラス21

10

20

30

40

50

に包含されている。さらに、"AutoMobile" クラス 23 は "Passenger's Car" クラス 24 および "Truck" クラス 25 を包含していることもわかる。

【0025】

このような矩形ダイアグラムを用いた包含関係の表示は、次のように変形することができる。例えば、図 2 のような包含関係を表すダイアグラムの一部、例えば紙面右側の端部や、紙面下側の端部を切り欠く（省略）表示としてもよい。ユーザはこれらの切り欠き部分を容易に推測できることから、必ずしも包含関係が不明瞭とはならない。このような表示によれば表示ウインドウ上の占有領域を節約できる。なお、切り欠き表示の領域はマウス等によるウインドウバーへの操作にダイナミックに連動して調整できることが好ましい。

【0026】

階層型データベースの各クラスは、インスタンスを持つことができる。インスタンスは実例ともいう。階層構造表示部 1 は、当該クラスがインスタンスを持つか否かを上記包含関係とともに明示的に表示する。

【0027】

従来の階層構造表示では、ユーザに対してクラスの開閉の指針が示されているような表示例はない。つまり、クラスを上位から下位に向かって開いてみなければ、下位のクラスにおけるインスタンスの存在有無を確認できない。

【0028】

本実施形態では、図 2 中の傘マークのようなマークを付けることにより、クラスの開閉を行わなくとも、上位のクラスでインスタンスの存在が確認できる。また、カバンマークのようなマークを付けることにより、明確にインスタンスの存在が確認できる。このようなインスタンスの有無表示によれば、ユーザは絞り込み検索を行いやすくなる。

20

【0029】

図 2 の例において、傘マーク（例えば参照数字 30）は、下位に（子）クラスが存在することを表し、鞄マーク（例えば参照数字 32, 33）は、クラスにインスタンスが存在することを表している。傘マークが付与されていないクラス（例えば "Sedan" クラス 29 や "Truck" クラス 25）は、末端クラスであり、すなわち下位クラスを持たない。

【0030】

閉じた傘マーク（例えば参照数字 31）は、その下位のクラス階層を開いていない状態を表す。この閉じた傘マーク 31 や 34 は、傘の色が無色（或いは白抜き）となっており、その下位クラスにインスタンスが存在しないことを明示している。例えば、"AutoBicycle" クラス 22 以下には下位クラスが存在するが、傘マーク 31 が閉じているため表示されていないことがわかる。また、この "AutoBicycle" クラス 22 には、登録されているインスタンスはないことがわかる。

30

【0031】

閉じた傘マークで有色のもの（例えば参照数字 35）は、クラス階層を開いていない状態を表している。傘マークが有色であることは、そのクラス以下にインスタンスが存在することを明示的に表す。具体例としては、"Wagon" クラス 35 以下には下位クラスが存在するが、傘が閉じているため表示されていない。また、傘マーク 35 が有色であるから、この "Wagon" クラス 35 は下位クラスのいずれかにインスタンスが存在することが分かる。

40

【0032】

開いた傘マークで無色（白抜き）のもの（例えば参照数字 34）は、クラス階層を開いた状態を表し、かつそのクラス以下にはインスタンスが存在しないことを表す。例えば、"SportsCar" クラス 26 の一階層下に存在する "OpenCar" クラス 27 が表示されている。傘マークは無色であるから、当該クラス以下には登録されているインスタンスが存在しないことが分かる。

【0033】

開いた傘マークで有色のもの（例えば参照数字 30）は、クラス階層を開いた状態を表し

50

、かつそのクラス以下にインスタンスが存在することを表す。例えば、"V e h i c l e" クラス 2 1 の一階層下に存在する" A u t o B i c y c l e" クラス 3 1 、" A u t o M o b i l e" クラス 2 3 を確認できる。その下位クラスである" S e d a n" クラス 2 9 、" T r u c k" クラス 2 5 にはインスタンスが存在するため、インスタンスそのものを表す鞠マーク 3 2 、3 3 が表示されている。

【0 0 3 4】

上位クラスにおける下位クラスのインスタンスの有無判定は、例えば図 3 のフローチャートで表す処理手順に基づいて行うことができる。図 3 の処理手順では、各クラスの情報として、下位クラスでのインスタンス有無フラグを持たせることにしている。インスタンス有無フラグは、インスタンス新規作成の際に更新されるものとする。

10

【0 0 3 5】

図 3 はインスタンス有無フラグの更新処理手順を示すフローチャートである。

【0 0 3 6】

まず、新規にインスタンスが挿入されたクラス (c l a s s) をセットする (ステップ S 1)。このクラスについて、フラグ (s u b _ i n s _ f l g) が O N であるか否かをステップ S 2 において判定する。フラグが O N と判定された場合には、下位にクラスが存在することを示して終了する。フラグが O F F であると判定された場合には、下位にクラスが存在しない。ステップ S 4 においてこのフラグを O N にセットし、次にステップ S S 5 において上位クラスが存在するか否かを判定する。上位クラスが存在しない場合は処理を終了する。上位クラスが存在する場合はステップ S 6 においてクラス (c l a s s) を上位クラスとしてステップ S 2 に戻る。この処理手順によれば、インスタンスの追加があるときにだけ、最小限の範囲でフラグの更新処理が行われるため、効率的である。図 2 のクラス階層構造表示は、このフラグが O N であるときに、傘マークの内部が有色となる。

20

【0 0 3 7】

図 4 は、階層構造表示に関するより具体的な G U I 画面の構成例を示す図である。図 4 において、クラスの包含関係はウインドウの領域 1 1 に示されている。この領域 1 1 内では、クラス間の包含関係を表すダイアグラムの右側端部と、上端部の表示が省略されている。

【0 0 3 8】

図 1 に示した階層構造表示装置の働きにより、階層構造表示部 1 とクラス情報表示部 2 とが連動し、階層構造表示部とプロパティリスト表示部 3 とが連動し、また、プロパティリスト表示部 3 とプロパティ情報表示部 4 とが連動し、表示内容が変更される。階層構造表示部 1 で選択されたクラスの名称、シノニム、定義などの情報をクラス情報表示部 2 により確認することができる。これは図 4 の領域 1 2 に対応する。また、階層構造表示部 1 で選択されたクラスで定義／継承されているプロパティのリストをプロパティリスト表示部 3 で確認することができる。これは、図 4 の領域 1 3 に対応する。このプロパティリスト表示部 3 で選択されたプロパティの情報は、プロパティ情報表示部 4 により確認することができる。プロパティの名称、シノニム、定義、データ型などの詳細情報が分かる。プロパティ情報表示部 4 による詳細情報の表示例を図 5 に示す。

30

【0 0 3 9】

「ヘリテージ (Her i t a g e) 表示」

40

図 6 は、選択されたクラスから上位のクラスへ向かって直系の親クラス 4 0 だけを表示している例である。このような表示をヘリテージ (Her i t a g e) 表示と呼ぶ。この例では、"ノート P C" クラス 4 1 から見た直系に相当するクラス間の親子関係が表現されている。

【0 0 4 0】

「検索開始ポイント (T r a v e r s a l 検索) の限定」

図 7 は検索開始ポイントを限定するトラバーサル (T r a v e r s a l) 検索を説明するための図である。階層型データベースでは、通常、階層にまたがったインスタンスの検索 (以降、T r a v e r s a l 検索と呼ぶ) を、任意のクラスから行うことができる。これ

50

は、階層の上位から検索をかけることにより、どのクラスのインスタンスかをユーザが意識することなく検索が行えるという利点がある。しかしながら、検索を開始するポイントが階層の上位であればあるほど、検索対象となるクラス、およびインスタンスが爆発的に増大する。このために検索時間がかかり、システムのパフォーマンスを低下させる恐れがある。そこで図7のように検索開始ポイントを限定する。これによれば膨大なデータに対する検索を回避できるようになり、システムのパフォーマンスの低下を防ぐことができる。

【0041】

検索開始ポイントは、インスタンスを持つクラスに限定することもできる。インスタンスは多くの場合、ツリーの末端近くのクラスに存在するため、検索対象クラスを少なく抑えることができる。図7の例で言うと、"Passenger Car"、"Sedan"、"Truck"の鞆マークが付与された三つのクラス24、29、25がトラバーサル検索の対象となる。

10

【0042】

また、システムアドミニストレータや辞書設計者などが、自クラスにインスタンスを含まなくとも、下位にインスタンスを含むようなクラスに対して、データベースや検索の負荷を考慮して、人的に設定することも可能である。図7の例でいうと、"Auto Bicycle"、"Sports Car"、"Open Car"など無色の鞆マークまたは、末端のクラスで鞆マークがついていないクラスに関しては、トラバーサル検索の対象とはならないが、他についてはその対象として、人的に検索開始ポイントを設定することが可能である。検索開始ポイントは、明示的にマークを挿入することでユーザに表示する。図7の"Auto Mobile"クラスに付与されているルーペマーク71がこれに相当する。"Auto Mobile"クラスからトラバーサル検索を行った場合、"Passenger Car"と"Sedan"、"Wagon"から以下、および"Truck"の各クラスに対して、検索条件に合ったインスタンスの検索を行うことができる。

20

【0043】

また、図8の処理手順に従い、検索開始ポイントを自動的に設定することも可能である。図8の処理手順では、自クラスにはインスタンスを含まないが、直下の子クラスの2箇所以上にインスタンスが存在する場合に、当該クラスを検索開始ポイントとする。検索開始ポイントの設定は、例えば、クラスの情報として各クラスに持たせ、インスタンスを挿入するタイミングで行なう。2つ以上の直下の子クラスにインスタンスが存在する場合、これを検索開始ポイントとする。

30

【0044】

`class. chld_ins`は、クラス(`class`)の直下の子クラスの、インスタンスの有無を示す変数である。初期値は、`NONE`(直下の子クラスにインスタンスが全くない)とする。一つの直下の子クラスにだけインスタンスが存在する場合には、該変数を`ONE`とし(ステップS7)、2つ以上の直下の子クラスにインスタンスが存在する場合には、`MULT`となる(ステップS6)。これが、`MULT`(複数)の場合、開始ポイントとする。図7の例では、"Auto Mobile"クラスは、直下のクラスである"Passenger Car"と"Truck"にインスタンスが存在するため、図8の処理手順に従って検索開始ポイントとして設定される。

40

【0045】

「部分継承表示」

図9は、マルチインヘリタンスを伴う階層構造の一例を示す図である。例えば、ISO13584では、単純木の階層構造のみを規定しており、マルチインヘリタンス(多重継承)は規定されていない。しかしながら、マルチインヘリタンスと類似するものとして、別のクラスの一部を継承することができるケースオブ(`CaseOf`)なる概念が知られている。これは、別のクラスで定義されたプロパティリストの一部を継承することができるというものである。

50

【0046】

図9の階層構造例では、実線が通常の継承関係を表し、破線90がケイスオブ（Case Of）による部分継承を表している。“Hybrid Vehicle”クラスは、“Electric Vehicle”クラスの子クラスであるが、ケイスオブ（Case Of）による部分継承によって同時に“Sedan”クラスのプロパティを継承している。

【0047】

図10は、階層構造中の部分継承を明示的に示す表示例である。ここでは、“Hybrid Vehicle”クラス100を“Electric Vehicle”クラス101の子クラスとしてのみ表示している。この表示例では、“Sedan”クラス100の下位には子クラスは存在しないから、部分継承しているクラスが存在するのどうかが分からぬ。しかし、“Hybrid Vehicle”クラス100が別のクラスから部分継承しているクラスであることを明示するために、他のクラスとは異なる表記（ここではクラス名称を例えれば斜字体表記としている）としている。

10

【0048】

一方、図11は図10とは異なる部分継承の他の表示例を示したものである。この例では、“Hybrid Vehicle”クラス110を“Sedan”クラス111の子クラスのように表示している。“Sedan”クラス111よりも下位のクラスである“Hybrid Vehicle”クラス110は、“Sedan”クラス111の正規の子クラスではなく、ケイスオブ（Case Of）により部分継承されている。このことを明示的に示すために、参照（Ref）マーク112を付与している。また、階層構造表示部1は、このRefマーク付のクラスが選択された場合、正規のクラスへジャンプする機構を備えており、ユーザは、“Sedan”クラス111を起点としても検索を行っていくことができる。

20

【0049】

「色別表示」

図12は、階層構造表示における色別表示を説明するための図である。階層構造表示部1により表示される階層構造の各階層の色と、プロパティリスト表示部3におけるプロパティリストの先頭マークの色とを同色で表示する。このような色別表示によれば、現在の選択クラス121からみた上位クラスのどの階層でプロパティリスト中の各プロパティが定義されているのかを、一目で確認することができ、極めて良好な視覚効果が得られる。

30

【0050】

「部分継承元クラス参照表示」

図13は、部分継承先のクラスから、部分継承したプロパティの出所クラスへの参照表示を示す図である。図13において、参照数字21は階層構造表示、22はプロパティリスト表示、23はプロパティ情報表示の例である。階層構造表示21における“Hybrid Vehicle”クラスおよびプロパティリスト表示22における“engine_type”プロパティは、ユーザによって選択されている項目であり、ハイライト表示されている。

40

【0051】

また、プロパティリスト表示22において表示されているプロパティの左端に示すCマーク25は、このプロパティが部分継承に係るものであることを明示するためのマークである。ユーザは、この“Hybrid Vehicle”クラスが部分継承している“engine_type”プロパティの詳細情報をプロパティ情報表示23を参照することで、このプロパティは“Sedan”クラスから継承されたものであることを確認できる。ここで、Jump（ジャンプ）ボタン24をユーザが押すと、階層構造表示21においては部分継承したプロパティの出所クラス“Sedan”クラスにフォーカスが切り替わり、これと同時にプロパティリスト表示22におけるプロパティリストが“Sedan”クラスのプロパティリストに表示更新される。

【0052】

「バーチャル（仮想）・ルート表示」

50

階層構造表示 21 は、通常、一つの階層構造を表示する。これは、上位クラスを持たないルート (R O O T) を源とするツリーに相当する。良く知られている C++ や Java 言語などに設けられたグラフィックライブラリなどでは、複数ツリー表示は、画面の切り替えによって行われる。このため、そのうちの一つのツリーしか一度に表示することができない。一方、本発明の実施形態では、複数のツリーに対してバーチャル・ルートを設け、それに対応するクラスを作成することにより、作成したバーチャルルートクラスの下位クラスとして、それぞれのツリーのルートを表現する。

【0053】

図 14 は、階層構造表示部におけるバーチャル・ルートを設けた場合の表示例を示している。バーチャル・ルート 140 は通常のクラスと同等に扱われる。これまでに説明した本実施形態に係わるすべての機能を、バーチャル・ルート 140 下においても通常クラスと同様に利用することができる。

10

【0054】

例えば、図 15 は、バーチャル・ルート 150 の一階層下の各ツリーのルート 151, 152 のみを初期表示するよう設定したものである。

【0055】

図 16 は、初期展開クラスの設定例を示す図である。ここでは、デフォルトの展開階数は「TREE_OPEN = 6」と設定されている。また、初期時に展開表示する個々のクラスは、「TREE_OPEN_CLASS」以下に、クラスの識別子 160 を列挙することで指定することができる。クラスの展開は、デフォルトの展開階数を原則とするが、このようにクラスの指定があれば、その枝部分については、該クラスまでの展開とする。

20

【0056】

以上説明した本発明の実施形態によれば、階層構造そのものについての情報（包含関係）のみならず、インスタンスの有無やプロパティリストなどを含むクラスに関する情報、プロパティに関する情報を効果的に提示するよう表示部に配置する。したがって、ユーザが操作しやすいインターフェースを提供できる。たとえば、階層表示するクラス範囲を、ユーザあるいは提供者が意図的により指定して限定することもでき、複雑な階層構造を必要な部分だけに絞ってユーザに提供することができる。また、検索開始ポイントを限定してユーザに提供することで、システムに負荷がかからない検索を行わせることもできる。

30

【0057】

なお、本発明は上述した実施形態に限定されず種々変形して実施可能である。例えば、本発明はクラス間に継承が発生するような階層型データベースのみならず、種々のシステムにおける階層構造の表示にも適用可能である。例えば、ファイルシステムにおけるディレクトリ構造ツリーの表示にも本発明は適用可能である。例えば、ディレクトリ毎に、そのインスタンスの存在有無を明示することなどである。

【0058】

【発明の効果】

以上説明したように、本発明によれば、階層型データベースにおけるクラス間の包含関係を適切かつ効果的に表示することのできる階層構造表示装置および方法を提供できる。

40

【図面の簡単な説明】

【図 1】 本発明の一実施形態に係る階層構造表示システムの概略構成を示すブロック図

【図 2】 上記実施形態に係るクラス間の包含関係を示す図

【図 3】 上記実施形態に係る下位クラスにおけるインスタンスの有無判定の処理手順の一例を示すフローチャート

【図 4】 上記実施形態に係る階層構造表示に関するより具体的な G U I 画面の構成例を示す図

【図 5】 上記実施形態に係るプロパティ情報表示部による詳細情報の表示例を示す図

【図 6】 上記実施形態に係るヘリテージ (Heritage) 表示の一例を示す図

【図 7】 上記実施形態に係るトラバーサル (Traversal) 検索を説明するための図

50

【図 8】上記実施形態に係る検索開始ポイントを自動的に設定する処理手順の一例を示すフローチャート

【図 9】上記実施形態に係るマルチインヘリタンスを伴う階層構造の一例を示す図

【図 10】上記実施形態に係る階層構造中の部分継承の表示例を示す図

【図 11】上記実施形態に係る部分継承の他の表示例を示す図

【図 12】上記実施形態に係る色別表示を説明するための図

【図 13】上記実施形態に係る部分継承先のクラスから、部分継承したプロパティの出所クラスへの参照表示を示す図

【図 14】上記実施形態に係る階層構造表示部におけるバーチャル・ルートを設けた場合の表示例を示す図

10

【図 15】上記実施形態に係るバーチャル・ルート設定時の初期表示例を示す図

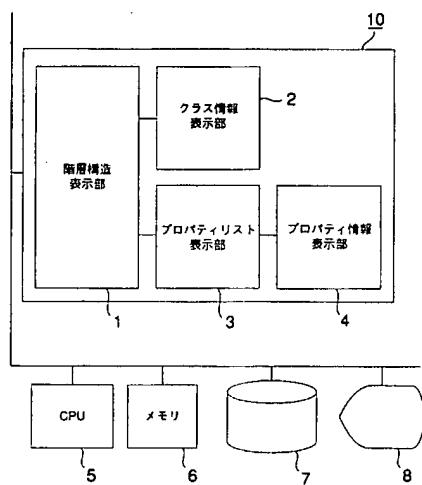
【図 16】上記実施形態に係る初期展開クラスの設定例を示す図

【符号の説明】

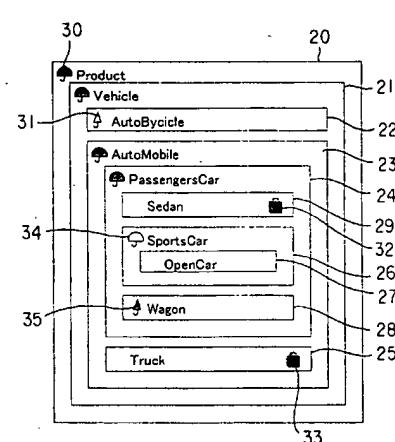
- 1 … 階層構造表示部
- 2 … クラス情報表示部
- 3 … プロパティリスト表示部
- 4 … プロパティ情報表示部
- 5 … C P U
- 6 … メモリ
- 7 … 二次記憶装置
- 8 … ディスプレイ

20

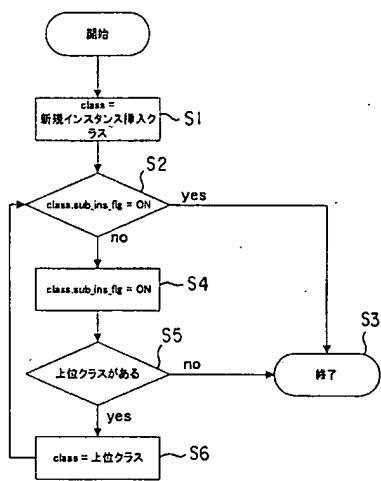
【図 1】



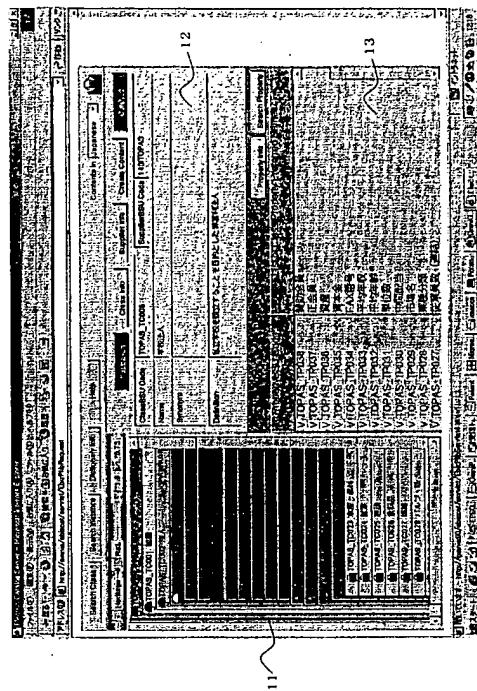
【図 2】



【図3】



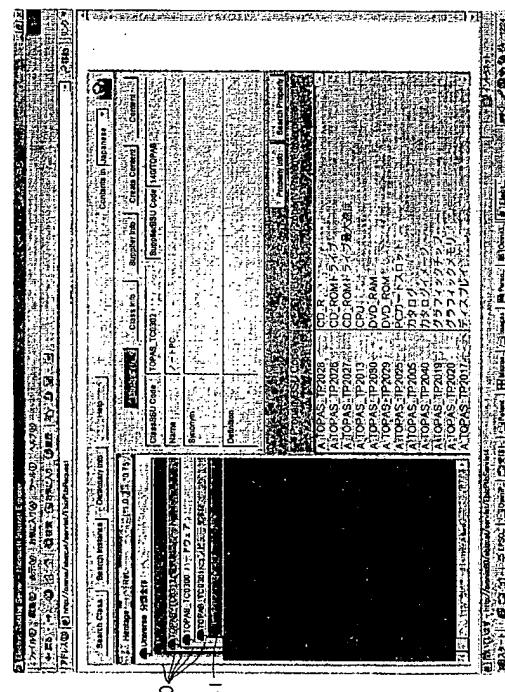
【図4】



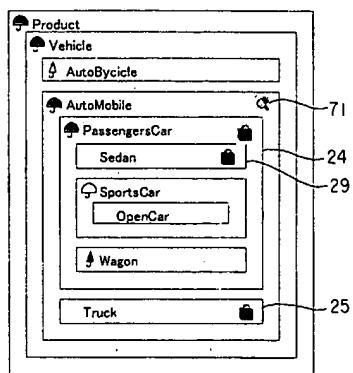
〔四〕 5



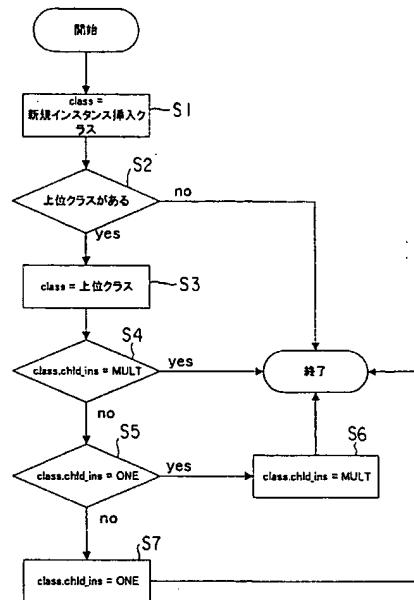
〔図6〕



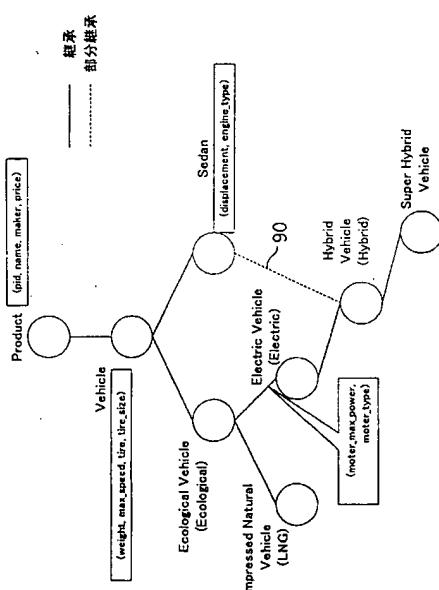
【図 7】



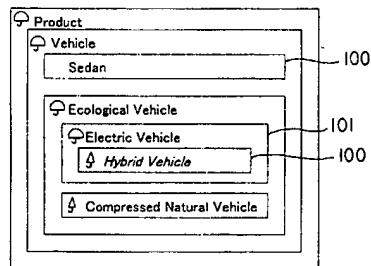
【図 8】



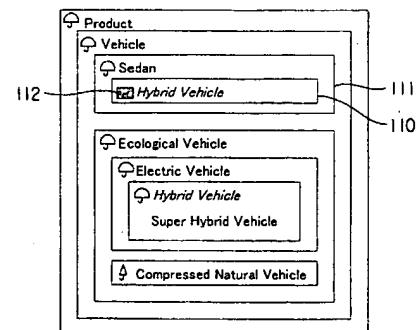
【図 9】



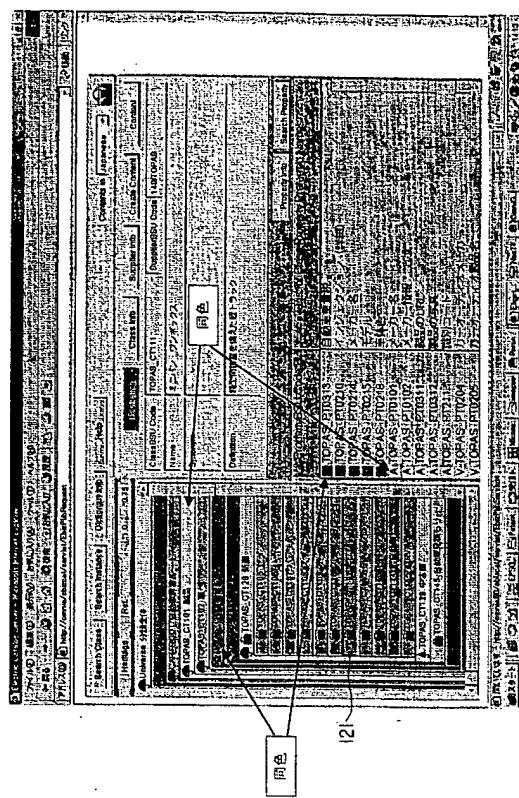
【図 10】



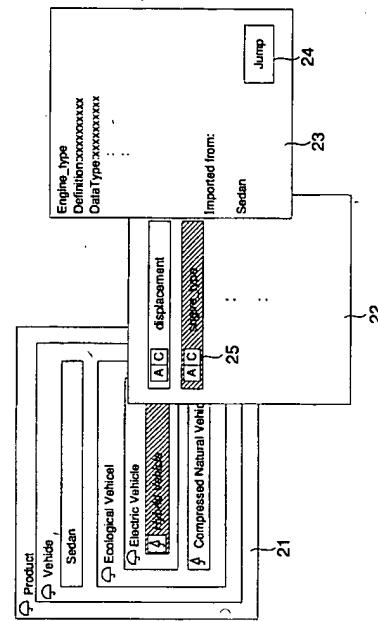
【図 11】



【図12】



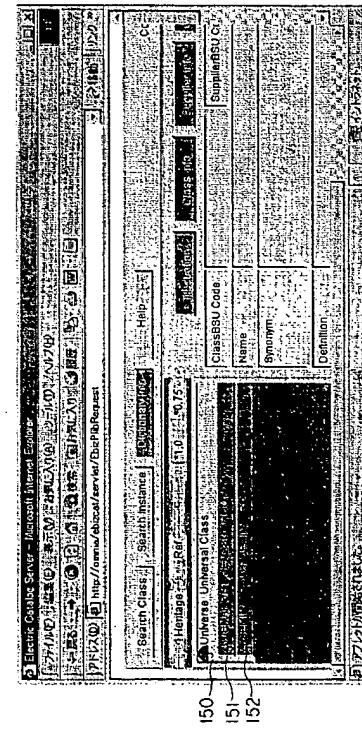
【図13】



【図14】



【図15】



【図 16】

```
TREE_OPEN=6
TREE_OPEN_CLASS
 140/TOPAS.TOPAS_CT101
 140/TOPAS.TOPAS_CT103
 140/TOPAS.TOPAS_CT104
}
END
```

フロントページの続き

(74)代理人 100070437

弁理士 河井 将次

(72)発明者 南野 典子

神奈川県川崎市幸区小向東芝町1番地 株式会社東芝研究開発センター内

(72)発明者 溝口 祐美子

神奈川県川崎市幸区小向東芝町1番地 株式会社東芝研究開発センター内

(72)発明者 村山 廣

神奈川県川崎市幸区小向東芝町1番地 株式会社東芝研究開発センター内

F ターム(参考) 5B082 GC02

* NOTICES *

JPO and INPIT are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

CLAIMS

[Claim(s)]

[Claim 1]

In a hierarchical structure display device of a hierarchical database with which a class which makes a layered structure was defined,

A hierarchical structure display device possessing a layered structure displaying means which displays said 1st and 2nd viewing areas so that at least a part of 1st viewing area that identifies any 1 class which makes said layered structure may contain all the 2nd viewing area that identifies a child class derived from this class.

[Claim 2]

The hierarchical structure display device possessing a class information displaying means which displays class information relevant to either said selected class of 1, and a child class according to claim 1.

[Claim 3]

The hierarchical structure display device possessing a property list displaying means which relates with said class information a list of properties which either said selected class of 1 and a child class have, and displays it according to claim 2.

[Claim 4]

The hierarchical structure display device possessing a property information displaying means which displays property information relevant to a property chosen from said property list according to claim 3.

[Claim 5]

The hierarchical structure display device comprising according to claim 3:
A class of a succeeding agency.

A means to match and display a property in said property list in a class of a succession place in the same color.

[Claim 6]

The hierarchical structure display device possessing a means changed according to switching operation to which display existence of the 2nd viewing area that identifies said child class was given to said class of 1 according to any one of claims 1 to 5.

[Claim 7]

The hierarchical structure display device possessing a means which indicates only the class which has an instance by deployment according to any one of claims 1 to 6.

[Claim 8]

The hierarchical structure display device possessing a means to display only a direct system class of a class of 1 selected into said layered structure according to any one of claims 1 to 7.

[Claim 9]

The hierarchical structure display device possessing a means to display a mark which identifies whether it relates with said any 1 class, and any one of said the child classes has an instance according to any one of claims 1 to 8.

[Claim 10]

A means to set any 1 class within said layered structure as search start points, A means to limit a class which sets up said search point to a class which has an

instance

The providing hierarchical structure display device according to any one of claims 1 to 9.

[Claim 11]

The hierarchical structure display device according to claim 10 limiting to a class which has at least two or more child classes which have an instance for a class which sets up said search point.

[Claim 12]

The hierarchical structure display device according to any one of claims 1 to 11, wherein said hierarchical database permits a multiple inheritance of a property from two or more parent classes.

[Claim 13]

The hierarchical structure display device possessing a means to refer to either of said parent classes which choose a property concerning said multiple inheritance and are equivalent to succession origin of this property according to claim 12.

[Claim 14]

The hierarchical structure display device comprising according to any one of claims 1 to 13:

The 1st layered structure that has the 1st route.

A means to set up a universal route common to the 2nd layered structure that has the 2nd route.

[Claim 15]

In the layered structure method of presentation of a hierarchical database with which a class which makes a layered structure was defined,

A layered structure displaying step which displays said 1st and 2nd viewing areas so that at least a part of 1st viewing area that identifies any 1 class which makes said layered structure may contain all the 2nd viewing area that identifies a child class derived from this class,

A step which receives selection of either said class of 1, and a child class,

A class information displaying step which is interlocked with said selection and displays class information relevant to either said this chosen class of 1, and a child class,

A property list displaying step which relates a list of properties which are interlocked with said selection and either said this chosen class of 1 and a child class have with said class information, and displays it

The providing layered structure method of presentation.

[Translation done.]

* NOTICES *

JPO and INPIT are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

DETAILED DESCRIPTION

[Detailed Description of the Invention]

[0001]

[Field of the Invention]

This invention relates to the suitable hierarchical structure display device and method for a hierarchical database.

[0002]

[Description of the Prior Art]

the technical field of today and computer software -- more -- a user -- progress is accomplished every day that a FRIENDLY function should be provided. For example, so that operation (operation) in which a user moves to a desired directory in use of a file system, and a desired file is opened can be performed visually and intuitively, The graphic user interface (GUI) can display directory (folder) structure on tree form combining a line, an icon, etc., and can show it to a user. In general-purpose operating systems, such as the operating systems (OS) (trademark) Windows (trademark), UNIX (trademark), and LINUX of Microsoft Corp., such a GUI function is adopted widely.

[0003]

About between each node (directory) of a directory structure tree, there are no relations between the information, including file etc., included in the node of a higher rank and the information included in a low-ranking node, such as inclusion or a subset. That is, each node which forms a directory structure tree and which begins from a root node only expresses the interconnection relation of the container which dedicates information, including a file etc.

[0004]

On the other hand, the database which makes representation an object oriented database (OODB) and an Object Relational Database (ORDB) has a layered structure, and has a mechanism in which a subdivided classification inherits the attribute of an upper classification in this layered structure. In such a database, there is the feature that an attribute accumulates by low-ranking classification by succession. It is also called "inheritance" that a subdivided classification inherits the attribute of an upper classification, and such art is indicated in much literature (for example, refer to the following nonpatent literature 1.).

[0005]

In an object oriented database (OODB)-related technical field, the classification in a hierarchy is called a "class" in many cases. In this specification, a "classification" and a "class" are used as a term which has the almost same meaning. In an object relational database (ORDB), the table which allowed succession is equivalent to a class. Between the tables in the hierarchical order, an attribute is inherited from the table of a higher rank to a low-ranking table. An attribute here is equivalent to the header information of the column which constitutes a higher rank table in ORDB, and this is inherited to a low order table.

[0006]

In this specification, a "hierarchical database" is called including both an object oriented database (OODB) and an Object Relational Database (ORDB). Data with the same attribute kind belonging to each hierarchy's class is called an "instance", and the set is called "population" of data.

[0007]

Thus, an above-mentioned directory structure tree differs in the character clearly

detailed description(2004-178015)

from the layered structure of a hierarchical database having the inclusion relation (succession relation) between classes.

[0008]

The service which applied the hierarchical database to the product catalog data base is already provided partly (see the following nonpatent literatures 2 and 3.).

[0009]

[Nonpatent literature 1]

Object-Oriented Concepts, Databases, and Applications, Edited by Won Kim, 1989, ACM Press

[0010]

[Nonpatent literature 2]

An incorporated company portal, the Internet <URL:<http://www.portalcorp.net>>

[0011]

[Nonpatent literature 3]

e-ingBiz.com, the Internet <URL:[http://www.e-ingbiz.com//CATALOG2 /servlet/CatalogSearch?Lang=ja](http://www.e-ingbiz.com//CATALOG2/servlet/CatalogSearch?Lang=ja)>

[0012]

[Problem(s) to be Solved by the Invention]

with the Explorer of a display [in / about a layered structure display / the conventional directory structure tree] Corp., for example, Microsoft, (trademark) - Indicate the low-ranking folder (directory) by deployment according to the opening-and-closing demand to the folder of a higher rank.

- By stage lowering, the folder twist expresses that it is a low-ranking folder to the folder of a higher rank.

- Make it by arranging stage lowering with every [of a hierarchy] level (depth) show easily that a level is the same folder.

what was said can be performed.

[0013]

However, these enable it to follow the connecting relation of a directory (node) to the last.

[0014]

Therefore, such a conventional layered structure display is not suitable for the hierarchical database that a property (attribute) is inherited from the class of a higher rank to a low-ranking class.

[0015]

The layered structure display in a hierarchical database is urged to be able to express the inclusion relation between classes in case each class expresses the concept characterized with the property to own. When the layered structure of a hierarchical database is constituted by the class, it is preferred that the existence of the instance of this class can be displayed.

[0016]

This invention is made in consideration of this situation, and is a thing. the purpose is to provide the hierarchical structure display device and method of displaying the inclusion relation between the classes which can be boiled and set appropriately and effectively.

[0017]

[Means for Solving the Problem]

In a hierarchical structure display device of a hierarchical database with which a class in which a hierarchical structure display device concerning this invention makes a layered structure was defined, A layered structure displaying means which displays said 1st and 2nd viewing areas so that at least a part of 1st viewing area that identifies any 1 class which makes said layered structure may contain all the 2nd viewing area that identifies a child class derived from this class is provided.

[0018]

The layered structure method of presentation which this invention requires for this invention is characterized by that the layered structure method of presentation of a hierarchical database with which a class which makes a layered structure was defined comprises:

A layered structure displaying step which displays said 1st and 2nd viewing areas so that at least a part of 1st viewing area that identifies any 1 class which makes said layered structure may contain all the 2nd viewing area that identifies a child class derived from this class.

detailed description(2004-178015)

A step which receives selection of either said class of 1, and a child class. A class information displaying step which is interlocked with said selection and displays class information relevant to either said this chosen class of 1, and a child class.

A property list displaying step which relates a list of properties which are interlocked with said selection and either said this chosen class of 1 and a child class have with said class information, and displays it.

[0019]

[Embodiment of the Invention]

Hereafter, the embodiment of this invention is described, referring to drawings.

[0020]

Drawing 1 is a block diagram showing the outline composition of the layered structure display system concerning one embodiment of this invention. A user depends neither search nor database management on the scale, detailed degree, etc., but enables it to perform this system appropriately and easily based on the layered structure display of a hierarchical database. The hierarchical structure display device concerning this invention is realizable as the application program 10 which operates on the computer using CPU5, the memory 6, the secondary memory 7, and the general-purpose computer provided with the basic hardware of display 8 grade. In this case, while providing the computer with the execution environment of the application program 10, the operating system for controlling hardware is also introduced. The application program 10 of a hierarchical structure display device is built on an operating system, and constitutes the graphical user interface (GUI) which collaborates with the display 8, the mouse which is not illustrated, a keyboard, etc. The hierarchical database is introduced into the secondary memory 7, for example.

[0021]

The hierarchical display is constituted by the layered structure indicator 1, the class information indicator 2, the property list indicator 3, and the property information indicator 4 as shown in the figure.

[0022]

In the hierarchical database that a low rank class inherits the class characteristics, such as a property, from a higher rank class, the way where the layered structure was expressed as not a relation but the inclusion relation of the mere higher rank in a hierarchy and a low rank like the regulation to ISO13584, For a user, the relation between classes is intelligible.

[0023]

Then, the layered structure indicator 1 displays the inclusion relation between classes clearly, as shown in drawing 2. Specifically, a class is displayed using diagrams, such as a rectangle, so that a higher rank class may include a low rank class.

[0024]

According to the display example of drawing 2, it is specified that the "Product" class 20 includes the "Vehicle" class 21. It turns out that this "Vehicle" class 21 contains two classes, the "AutoBicycle" class 22 and the "AutoMobile" class 23. In other words, two classes, the "AutoBicycle" class 22 and the "AutoMobile" class 23, are included by the "Vehicle" class 21. It also turns out that the "AutoMobile" class 23 includes the "PassengerCar" class 24 and the "Truck" class 25.

[0025]

The display of the inclusion relation using such a rectangle diagram can be transformed as follows. For example, it is good also as a display which cuts and lacks a part of diagram showing inclusion relation like drawing 2, for example, the end on the right-hand side of space, and the end of the space bottom (abbreviation). A user does not necessarily become ambiguous [inclusion relation] from the ability of these notching portions to be guessed easily. According to such a display, the occupied area on a display window can be saved. As for the field of a notching display, it is preferred that it interlocks dynamically and can adjust to the operation to the window bar by a mouse etc.

[0026]

Each class of a hierarchical database can have an instance. An instance also calls it an example. The layered structure indicator 1 indicates whether the class concerned has an instance clearly with the above-mentioned inclusion relation.

detailed description(2004-178015)

[0027]

There is no display example that the indicator of opening and closing of a class is shown to the user by the conventional layered structure display. That is, if a class is not opened toward a low rank from a higher rank, existence of the instance in a low-ranking class cannot be checked.

[0028]

At this embodiment, even if it does not open and close a class by attaching a mark like the umbrella mark in drawing 2, existence of an instance can be checked in the class of a higher rank. Existence of an instance can be clearly checked by attaching a mark like a bag mark. According to the existence display of such an instance, it becomes easy for a user to perform narrowing retrieval.

[0029]

In the example of drawing 2, an umbrella mark (for example, reference number 30) means that a class (child) exists in a low rank, and the bag mark (for example, reference numbers 32 and 33) means that an instance exists in a class. The class (for example, the "Sedan" class 29 and the "Truck" class 25) to which the umbrella mark is not given is an end class, namely, does not have a low rank class.

[0030]

The closed umbrella mark (for example, reference number 31) expresses the state where the class hierarchy of the low rank is not opened. The color of the umbrella is colorless (or white) and this closed umbrella mark 31 and 34 specify that an instance does not exist in that low rank class. For example, although a low rank class exists in 22 or less "AutoBicycle" class, it turns out that it is not displayed since the umbrella mark 31 has closed. It turns out that there is no instance registered in this "AutoBicycle" class 22.

[0031]

The colored thing (for example, reference number 35) expresses with the closed umbrella mark the state where the class hierarchy is not opened. That an umbrella mark is colored means clearly that an instance exists below in the class. As an example, although a low rank class exists in 35 or less "Wagon" class, since the umbrella has closed, it is not displayed. As for it, since the umbrella mark 35 is colored, this "Wagon" class 35 turns out that an instance exists in either of the low rank classes.

[0032]

A colorless (white) thing (for example, reference number 34) expresses with the open umbrella mark the state where the class hierarchy was opened, and it means that an instance does not exist to below the class. For example, the "OpenCar" class 27 which exists under the ground-floor layer of the "Sportscar" class 26 is displayed. Since an umbrella mark is colorless, it turns out that the instance registered does not exist below in the class concerned.

[0033]

It means that a colored thing (for example, reference number 30) expresses with the open umbrella mark the state where the class hierarchy was opened, and an instance exists below in the class. For example, the "AutoBicycle" class 31 and the "AutoMobile" class 23 which exist under the ground-floor layer of the "Vehicle" class 21 can be checked. Since an instance exists in the "Sedan" class 29 and the "Truck" class 25 which are the low rank class, the bag marks 32 and 33 showing the instance itself are displayed.

[0034]

The existence judging of the instance of the low rank class in a higher rank class can be performed based on the procedure expressed with the flow chart of drawing 3, for example. In the procedure of drawing 3, the instance existence flag in the low rank class is given as information on each class. An instance existence flag shall be updated in the case of instance new production.

[0035]

Drawing 3 is a flow chart which shows the update process procedure of an instance existence flag.

[0036]

First, the class (class) in which the instance was inserted newly is set (Step S1). About this class, it is judged in Step S2 whether a flag (sub_ins_flg) is ON. When a flag is judged to be ON, it is shown that a class exists in a low rank and it ends. When judged with a flag being OFF, a class does not exist in a low rank. This flag is set to ON in step S4, and it is judged whether next in step S5, a higher rank

detailed description(2004-178015)

class exists. Processing is ended when a higher rank class does not exist. When a higher rank class exists, in Step S6, it returns to Step S2 by making class (class) into a higher rank class. Since according to this procedure the update process of a flag is performed in the minimum range only when there is an addition of an instance, it is efficient. In the class hierarchy structure display of drawing 2, when this flag is ON, the inside of an umbrella mark becomes colored.

[0037]

Drawing 4 is a figure showing the example of composition of the more concrete GUI picture about a layered structure display. In drawing 4, the inclusion relation of the class is shown in the field 11 of the window. In this field 11, the display of the right side end part of the diagram showing the inclusion relation between classes and an upper bed part is omitted.

[0038]

By work of the hierarchical structure display device shown in drawing 1, the layered structure indicator 1 and the class information indicator 2 interlock, and a layered structure indicator and the property list indicator 3 interlock, and the property list indicator 3 and the property information indicator 4 interlock, and display information is changed. Information, including the name of the class selected by the layered structure indicator 1, synonym, a definition, etc., can be checked by the class information indicator 2. This corresponds to the field 12 of drawing 4. The list of properties defined / inherited in the class selected by the layered structure indicator 1 can be checked by the property list indicator 3. This corresponds to the field 13 of drawing 4. The information on the property selected by this property list indicator 3 can be checked by the property information indicator 4. The detailed information of the name of a property, synonym, a definition, a data type, etc. is known. The display example of the detailed information by the property information indicator 4 is shown in drawing 5.

[0039]

"Heritage (Heritage) display"

Drawing 6 is an example which shows only the direct parent class 40 toward the class of a higher rank from the selected class. Such a display is called a heritage (Heritage) display. In this example, the child-parent relationship between the classes equivalent to the direct system seen from the "notebook PC" class 41 is expressed.

[0040]

"Limitation of search start points (Traversal search)"

Drawing 7 is a figure for explaining the traversal (Traversal) search which limits search start points. The instance over a hierarchy can usually be searched with a hierarchical database from arbitrary classes (it is henceforth called Traversal search). This has the advantage that it can search without a user being conscious of of which class it is an instance, by performing a search from a hierarchy's higher rank. However, if the point which starts search is a hierarchy's higher rank, in a certain forge fire, the class used as a retrieval object and an instance will increase explosively. For this reason, search time starts and there is a possibility of reducing the performance of a system. Then, search start points are limited like drawing 7. According to this, the search to a vast quantity of data can be avoided now, and the fall of the performance of a system can be prevented.

[0041]

Search start points can also be limited to a class with an instance. Since an instance exists in the class near the end of a tree in many cases, it can stop a retrieval object class few. If it says in the example of drawing 7, the three classes 24, 29, and 25 to which "Passenger Car", "Sedan", and the bag mark of "Truck" were given will be the targets of traversal search.

[0042]

He is able to set up human in consideration of a database or the load of search to a class which contains an instance in a low rank, even if neither a system administrator nor a dictionary designer contains an instance in a self-class. when it says in the example of drawing 7, colorless umbrella marks, such as "Auto Bicycle", "Sports Car", and "Open Car". Or although it is not the target of traversal search about the class for which it is not having the bag mark in the class of an end, about others, it is possible as the object to set up search start points human. Search start points are displayed on a user by inserting a mark clearly. The magnifying glass mark 71 given to the "Auto Mobile" class of drawing 7

detailed description(2004-178015)

is equivalent to this. When traversal search is performed from a "Auto Mobile" class, the instance as for which the following suited the search condition from "Passenger Car", "Sedan", and "Wagon" to each class of "Truck" can be searched.

[0043]

It is also possible to set up search start points automatically according to the procedure of drawing 8. In the procedure of drawing 8, although an instance is not included in a self-class, when an instance exists in two or more places of a child class [directly under], let the class concerned be search start points. Setting out of search start points is given to each class as information on a class, and is performed to the timing which inserts an instance, for example. When an instance exists in two or more child classes [directly under], let this be search start points.

[0044]

class.chld_ins is a variable which shows the existence of an instance of the child class [directly under] of class (class). An initial value is set to NONE (there is no instance in a child class [directly under]). When an instance exists only in one child class [directly under], this variable is set to ONE (Step S7), and it is set to MULT when an instance exists in two or more child classes [directly under] (Step S6). When this is MULT (plurality), it is considered as start points. In the example of drawing 7, since an instance exists in "Passenger Car" and "Truck" which are classes [directly under], a "Auto Mobile" class is set up as search start points according to the procedure of drawing 8.

[0045]

"Partial succession display"

Drawing 9 is a figure showing an example of the layered structure accompanied by multi-inheritance. For example, ISO13584 has prescribed only the layered structure of a simple tree and multi-inheritance (multiple inheritance) is not prescribed. However, the cay which can inherit a part of another class as multi-inheritance and a similar thing -- SUOBU (CaseOf) -- the concept is known. Some property lists defined by another class are inheritable for this.

[0046]

A solid line expresses the usual succession relation with the example of a layered structure of drawing 9, and the dashed line 90 expresses the partial succession by cay SUOBU (CaseOf) with it. Although a "Hybrid Vehicle" class is a child class of a "Electric Vehicle" class, it has inherited the property of a "Sedan" class simultaneously by partial succession by cay SUOBU (CaseOf).

[0047]

Drawing 10 is a display example which shows the partial succession in a layered structure clearly. Here, the "Hybrid Vehicle" class 100 is displayed only as a child class of the "Electric Vehicle" class 101. In this display example, since a child class does not exist in the low rank of the "Sedan" class 100, the class which is carrying out partial succession exists -- there is no telling how be. However, since the "Hybrid Vehicle" class 100 specifies that it is the class which is carrying out partial succession from another class, it is considered as a different notation (here, the class name is considered for example, as the italic type notation) from other classes.

[0048]

On the other hand, drawing 11 shows other display examples of the partial succession from which drawing 10 differs. In this example, the "Hybrid Vehicle" class 110 is displayed like the child class of the "Sedan" class 111. Partial succession of the "HybridVehicle" class 110 which is a low-ranking class is carried out by not a regular child class but cay SUOBU (CaseOf) of the "Sedan" class 111 rather than the "Sedan" class 111. In order to show this clearly, the reference (Ref) mark 112 is given. The layered structure indicator 1 is provided with the mechanism jumped to a regular class when a class with this Ref mark is chosen, and the user can search also considering the "Sedan" class 111 as a starting point.

[0049]

"A display according to color"

Drawing 12 is a figure for explaining the display according to color in a layered structure display. The color of each hierarchy of the layered structure displayed by the layered structure indicator 1 and the color of the top mark of the property list in the property list indicator 3 are displayed in the same color. According to such a display according to color, it can be checked whether each property in a property

detailed description(2004-178015)

list is defined by which hierarchy of the higher rank class seen from the present selected class 121 at a glance, and a very good visual effect is acquired.

[0050]

"Partial succession former class reference display"

Drawing 13 is a figure showing a reference display in the source class [class / of a partial succession place] of the property which carried out partial succession. As for the reference number 21, in drawing 13, a property list display and 23 are the examples of a property information display a layered structure display and 22. The "Hybrid Vehicle" class in the layered structure display 21 and the "engine_type" property in the property list display 22 are items chosen by the user, and highlighting is carried out.

[0051]

The C mark 25 shown in the left end of the property currently displayed in the property list display 22 is a mark for specifying that it is what requires this property for partial succession. A user is that this "Hybrid Vehicle" class refers to the property information display 23 for the detailed information of the "engine_type" property which is carrying out partial succession, and it can check that this property is inherited from a "Sedan" class. If a user pushes the Jump (jump) button 24, a focus will change to the source class "Sedan" class of the property which carried out partial succession in the layered structure display 21 here, Renewal of a display of the property list in the property list display 22 is carried out simultaneously with this at the property list of a "Sedan" class.

[0052]

"Virtual (imagination) and route table Shimesu"

The layered structure display 21 usually displays one layered structure. This is equivalent to the tree which makes a source route (ROOT) without a higher rank class. Two or more tree views are performed by the change of a screen in the graphic library established in C++ known well, a Java language, etc. For this reason, only one tree of them can be displayed at once. On the other hand, in the embodiment of this invention, the route of each tree is expressed as a low rank class of the created virtual route class by providing a virtual route to two or more trees, and creating the class corresponding to it.

[0053]

Drawing 14 shows the display example at the time of providing the virtual route in a layered structure indicator. The virtual route 140 is treated on a par with the usual class. Under the virtual route 140, all the functions concerning this embodiment described so far can usually be used like a class.

[0054]

For example, drawing 15 is set up display only the route 151,152 of each tree under the ground-floor layer of the virtual route 150 the first stage.

[0055]

Drawing 16 is a figure showing the example of setting out of an initial deployment class. Here, the default deployment number of stories is set to "TREE_OPEN=6." Each class which indicates by deployment at the time of the first stage can be specified below as "TREE_OPEN_CLASS" by enumerating the identifiers 160 of a class. Although carried out a principle [a default deployment number of stories], if deployment of a class has specification of a class in this way, it will be considered as deployment to this class about the branch parts.

[0056]

According to the embodiment of this invention explained above, it arranges to an indicator so that the information about the information about the class containing existence, a property list, etc. of not only the information (inclusion relation) about the layered structure itself but an instance and a property may be shown effectively. Therefore, the interface which a user tends to operate can be provided. For example, a user or a donor can specify intentionally the class range which carries out a hierarchical display more, it can also be limited, a complicated layered structure can be extracted only to a required portion, and it can provide for a user. The search which does not require load for a system can also be made to perform by limiting search start points and providing for a user.

[0057]

This invention is not limited to the embodiment mentioned above, but changes variously, and is feasible. For example, this invention is applicable not only to a hierarchical database which succession generates between classes but the display of

detailed description(2004-178015)

the layered structure in various systems. For example, this invention is applicable also to the display of the directory structure tree in a file system. For example, it is specifying the existence of the instance for every directory etc.

[0058]

[Effect of the Invention]

As explained above, according to this invention, the hierarchical structure display device and method of displaying appropriately and effectively the inclusion relation between the classes in a hierarchical database can be provided.

[Brief Description of the Drawings]

[Drawing 1]The block diagram showing the outline composition of the layered structure display system concerning one embodiment of this invention

[Drawing 2]The figure showing the inclusion relation between the classes concerning the above-mentioned embodiment

[Drawing 3]The flow chart which shows an example of the procedure of an existence judging of the instance in the low rank class concerning the above-mentioned embodiment

[Drawing 4]The figure showing the example of composition of the more concrete GUI picture about the layered structure display concerning the above-mentioned embodiment

[Drawing 5]The figure showing the display example of the detailed information by the property information indicator concerning the above-mentioned embodiment

[Drawing 6]The figure showing an example of the heritage (Heritage) display concerning the above-mentioned embodiment

[Drawing 7]The figure for explaining the traversal (Traversal) search concerning the above-mentioned embodiment

[Drawing 8]The flow chart which shows an example of procedure which sets up automatically the search start points concerning the above-mentioned embodiment

[Drawing 9]The figure showing an example of the layered structure accompanied by the multi-inheritance concerning the above-mentioned embodiment

[Drawing 10]The figure showing the display example of the partial succession in the layered structure concerning the above-mentioned embodiment

[Drawing 11]The figure showing other display examples of the partial succession concerning the above-mentioned embodiment

[Drawing 12]The figure for explaining the display according to color concerning the above-mentioned embodiment

[Drawing 13]The figure showing a reference display in the source class [class / of the partial succession place concerning the above-mentioned embodiment] of the property which carried out partial succession

[Drawing 14]The figure showing the display example at the time of providing the virtual route in the layered structure indicator concerning the above-mentioned embodiment

[Drawing 15]The figure showing the initial display example at the time of the virtual route setting concerning the above-mentioned embodiment

[Drawing 16]The figure showing the example of setting out of the initial deployment class concerning the above-mentioned embodiment

[Description of Notations]

- 1 -- Layered structure indicator
- 2 -- Class information indicator
- 3 -- Property list indicator
- 4 -- Property information indicator
- 5 -- CPU
- 6 -- Memory
- 7 -- Secondary memory
- 8 -- Display

[Translation done.]

* NOTICES *

JPO and INPIT are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

MEANS

[Means for Solving the Problem]

In a hierarchical structure display device of a hierarchical database with which a class in which a hierarchical structure display device concerning this invention makes a layered structure was defined, A layered structure displaying means which displays said 1st and 2nd viewing areas so that at least a part of 1st viewing area that identifies any 1 class which makes said layered structure may contain all the 2nd viewing area that identifies a child class derived from this class is provided.

[0018]

The layered structure method of presentation which this invention requires for this invention is characterized by that the layered structure method of presentation of a hierarchical database with which a class which makes a layered structure was defined comprises:

A layered structure displaying step which displays said 1st and 2nd viewing areas so that at least a part of 1st viewing area that identifies any 1 class which makes said layered structure may contain all the 2nd viewing area that identifies a child class derived from this class.

A step which receives selection of either said class of 1, and a child class.

A class information displaying step which is interlocked with said selection and displays class information relevant to either said this chosen class of 1, and a child class.

A property list displaying step which relates a list of properties which are interlocked with said selection and either said this chosen class of 1 and a child class have with said class information, and displays it.

[0019]

[Embodiment of the Invention]

Hereafter, the embodiment of this invention is described, referring to drawings.

[0020]

Drawing 1 is a block diagram showing the outline composition of the layered structure display system concerning one embodiment of this invention. A user depends neither search nor database management on the scale, detailed degree, etc., but enables it to perform this system appropriately and easily based on the layered structure display of a hierarchical database. The hierarchical structure display device concerning this invention is realizable as the application program 10 which operates on the computer using CPU5, the memory 6, the secondary memory 7, and the general-purpose computer provided with the basic hardware of display 8 grade. In this case, while providing the computer with the execution environment of the application program 10, the operating system for controlling hardware is also introduced. The application program 10 of a hierarchical structure display device is built on an operating system, and constitutes the graphical user interface (GUI) which collaborates with the display 8, the mouse which is not illustrated, a keyboard, etc. The hierarchical database is introduced into the secondary memory 7, for example.

[0021]

The hierarchical display is constituted by the layered structure indicator 1, the class information indicator 2, the property list indicator 3, and the property information indicator 4 as shown in the figure.

[0022]

means(2004-178015)

In the hierarchical database that a low rank class inherits the class characteristics, such as a property, from a higher rank class, the way where the layered structure was expressed as not a relation but the inclusion relation of the mere higher rank in a hierarchy and a low rank like the regulation to ISO13584, For a user, the relation between classes is intelligible.

[0023]

Then, the layered structure indicator 1 displays the inclusion relation between classes clearly, as shown in drawing 2. Specifically, a class is displayed using diagrams, such as a rectangle, so that a higher rank class may include a low rank class.

[0024]

According to the display example of drawing 2, it is specified that the "Product" class 20 includes the "Vehicle" class 21. It turns out that this "Vehicle" class 21 contains two classes, the "AutoBicycle" class 22 and the "AutoMobile" class 23. In other words, two classes, the "AutoBicycle" class 22 and the "AutoMobile" class 23, are included by the "Vehicle" class 21. It also turns out that the "AutoMobile" class 23 includes the "PassengerCar" class 24 and the "Truck" class 25.

[0025]

The display of the inclusion relation using such a rectangle diagram can be transformed as follows. For example, it is good also as a display which cuts and lacks a part of diagram showing inclusion relation like drawing 2, for example, the end on the right-hand side of space, and the end of the space bottom (abbreviation). A user does not necessarily become ambiguous [inclusion relation] from the ability of these notching portions to be guessed easily. According to such a display, the occupied area on a display window can be saved. As for the field of a notching display, it is preferred that it interlocks dynamically and can adjust to the operation to the window bar by a mouse etc.

[0026]

Each class of a hierarchical database can have an instance. An instance also calls it an example. The layered structure indicator 1 indicates whether the class concerned has an instance clearly with the above-mentioned inclusion relation.

[0027]

There is no display example that the indicator of opening and closing of a class is shown to the user by the conventional layered structure display. That is, if a class is not opened toward a low rank from a higher rank, existence of the instance in a low-ranking class cannot be checked.

[0028]

At this embodiment, even if it does not open and close a class by attaching a mark like the umbrella mark in drawing 2, existence of an instance can be checked in the class of a higher rank. Existence of an instance can be clearly checked by attaching a mark like a bag mark. According to the existence display of such an instance, it becomes easy for a user to perform narrowing retrieval.

[0029]

In the example of drawing 2, an umbrella mark (for example, reference number 30) means that a class (child) exists in a low rank, and the bag mark (for example, reference numbers 32 and 33) means that an instance exists in a class. The class (for example, the "Sedan" class 29 and the "Truck" class 25) to which the umbrella mark is not given is an end class, namely, does not have a low rank class.

[0030]

The closed umbrella mark (for example, reference number 31) expresses the state where the class hierarchy of the low rank is not opened. The color of the umbrella is colorless (or white) and this closed umbrella mark 31 and 34 specify that an instance does not exist in that low rank class. For example, although a low rank class exists in 22 or less "AutoBicycle" class, it turns out that it is not displayed since the umbrella mark 31 has closed. It turns out that there is no instance registered in this "AutoBicycle" class 22.

[0031]

The colored thing (for example, reference number 35) expresses with the closed umbrella mark the state where the class hierarchy is not opened. That an umbrella mark is colored means clearly that an instance exists below in the class. As an example, although a low rank class exists in 35 or less "Wagon" class, since the umbrella has closed, it is not displayed. As for it, since the umbrella mark 35 is colored, this "Wagon" class 35 turns out that an instance exists in either of the

low rank classes.

[0032]

A colorless (white) thing (for example, reference number 34) expresses with the open umbrella mark the state where the class hierarchy was opened, and it means that an instance does not exist to below the class. For example, the "OpenCar" class 27 which exists under the ground-floor layer of the "SportsCar" class 26 is displayed. Since an umbrella mark is colorless, it turns out that the instance registered does not exist below in the class concerned.

[0033]

It means that a colored thing (for example, reference number 30) expresses with the open umbrella mark the state where the class hierarchy was opened, and an instance exists below in the class. For example, the "AutoBicycle" class 31 and the "AutoMobile" class 23 which exist under the ground-floor layer of the "Vehicle" class 21 can be checked. Since an instance exists in the "Sedan" class 29 and the "Truck" class 25 which are the low rank class, the bag marks 32 and 33 showing the instance itself are displayed.

[0034]

The existence judging of the instance of the low rank class in a higher rank class can be performed based on the procedure expressed with the flow chart of drawing 3, for example. In the procedure of drawing 3, the instance existence flag in the low rank class is given as information on each class. An instance existence flag shall be updated in the case of instance new production.

[0035]

Drawing 3 is a flow chart which shows the update process procedure of an instance existence flag.

[0036]

First, the class (class) in which the instance was inserted newly is set (Step S1). About this class, it is judged in Step S2 whether a flag (sub_ins_flg) is ON. When a flag is judged to be ON, it is shown that a class exists in a low rank and it ends. When judged with a flag being OFF, a class does not exist in a low rank. This flag is set to ON in step S4, and it is judged whether next in step S5, a higher rank class exists. Processing is ended when a higher rank class does not exist. When a higher rank class exists, in Step S6, it returns to Step S2 by making class (class) into a higher rank class. Since according to this procedure the update process of a flag is performed in the minimum range only when there is an addition of an instance, it is efficient. In the class hierarchy structure display of drawing 2, when this flag is ON, the inside of an umbrella mark becomes colored.

[0037]

Drawing 4 is a figure showing the example of composition of the more concrete GUI picture about a layered structure display. In drawing 4, the inclusion relation of the class is shown in the field 11 of the window. In this field 11, the display of the right side end part of the diagram showing the inclusion relation between classes and an upper bed part is omitted.

[0038]

By work of the hierarchical structure display device shown in drawing 1, the layered structure indicator 1 and the class information indicator 2 interlock, and a layered structure indicator and the property list indicator 3 interlock, and the property list indicator 3 and the property information indicator 4 interlock, and display information is changed. Information, including the name of the class selected by the layered structure indicator 1, synonym, a definition, etc., can be checked by the class information indicator 2. This corresponds to the field 12 of drawing 4. The list of properties defined / inherited in the class selected by the layered structure indicator 1 can be checked by the property list indicator 3. This corresponds to the field 13 of drawing 4. The information on the property selected by this property list indicator 3 can be checked by the property information indicator 4. The detailed information of the name of a property, synonym, a definition, a data type, etc. is known. The display example of the detailed information by the property information indicator 4 is shown in drawing 5.

[0039]

"Heritage (Heritage) display"

Drawing 6 is an example which shows only the direct parent class 40 toward the class of a higher rank from the selected class. Such a display is called a heritage (Heritage) display. In this example, the child-parent relationship between the

classes equivalent to the direct system seen from the "notebook PC" class 41 is expressed.

[0040]

"Limitation of search start points (Traversal search)"

Drawing 7 is a figure for explaining the traversal (Traversal) search which limits search start points. The instance over a hierarchy can usually be searched with a hierarchical database from arbitrary classes (it is henceforth called Traversal search). This has the advantage that it can search without a user being conscious of of which class it is an instance, by performing a search from a hierarchy's higher rank. However, if the point which starts search is a hierarchy's higher rank, in a certain forge fire, the class used as a retrieval object and an instance will increase explosively. For this reason, search time starts and there is a possibility of reducing the performance of a system. Then, search start points are limited like drawing 7. According to this, the search to a vast quantity of data can be avoided now, and the fall of the performance of a system can be prevented.

[0041]

Search start points can also be limited to a class with an instance. Since an instance exists in the class near the end of a tree in many cases, it can stop a retrieval object class few. If it says in the example of drawing 7, the three classes 24, 29, and 25 to which "Passenger Car", "Sedan", and the bag mark of "Truck" were given will be the targets of traversal search.

[0042]

He is able to set up human in consideration of a database or the load of search to a class which contains an instance in a low rank, even if neither a system administrator nor a dictionary designer contains an instance in a self-class. When it says in the example of drawing 7, colorless umbrella marks, such as "Auto Bicycle", "Sports Car", and "Open Car". Or although it is not the target of traversal search about the class for which it is not having the bag mark in the class of an end, about others, it is possible as the object to set up search start points human. Search start points are displayed on a user by inserting a mark clearly. The magnifying glass mark 71 given to the "Auto Mobile" class of drawing 7 is equivalent to this. When traversal search is performed from a "Auto Mobile" class, the instance as for which the following suited the search condition from "Passenger Car", "Sedan", and "Wagon" to each class of "Truck" can be searched.

[0043]

It is also possible to set up search start points automatically according to the procedure of drawing 8. In the procedure of drawing 8, although an instance is not included in a self-class, when an instance exists in two or more places of a child class [directly under], let the class concerned be search start points. Setting out of search start points is given to each class as information on a class, and is performed to the timing which inserts an instance, for example. When an instance exists in two or more child classes [directly under], let this be search start points.

[0044]

class.chld_ins is a variable which shows the existence of an instance of the child class [directly under] of class (class). An initial value is set to NONE (there is no instance in a child class [directly under]). When an instance exists only in one child class [directly under], this variable is set to ONE (Step S7), and it is set to MULT when an instance exists in two or more child classes [directly under] (Step S6). When this is MULT (plurality), it is considered as start points. In the example of drawing 7, since an instance exists in "Passenger Car" and "Truck" which are classes [directly under], a "Auto Mobile" class is set up as search start points according to the procedure of drawing 8.

[0045]

"Partial succession display"

Drawing 9 is a figure showing an example of the layered structure accompanied by multi-inheritance. For example, ISO13584 has prescribed only the layered structure of a simple tree and multi-inheritance (multiple inheritance) is not prescribed. however, the cay which can inherit a part of another class as multi-inheritance and a similar thing -- SUOBU (CaseOf) -- the concept is known. Some property lists defined by another class are inheritable for this.

[0046]

A solid line expresses the usual succession relation with the example of a layered

means(2004-178015)

structure of drawing 9, and the dashed line 90 expresses the partial succession by cay SUOBU (CaseOf) with it. Although a "Hybrid Vehicle" class is a child class of a "Electric Vehicle" class, it has inherited the property of a "Sedan" class simultaneously by partial succession by cay SUOBU (CaseOf).

[0047]

Drawing 10 is a display example which shows the partial succession in a layered structure clearly. Here, the "Hybrid Vehicle" class 100 is displayed only as a child class of the "Electric Vehicle" class 101. in this display example, since a child class does not exist in the low rank of the "Sedan" class 100, the class which is carrying out partial succession exists -- there is no telling how be. However, since the "Hybrid Vehicle" class 100 specifies that it is the class which is carrying out partial succession from another class, it is considered as a different notation (here, the class name is considered for example, as the italic type notation) from other classes.

[0048]

On the other hand, drawing 11 shows other display examples of the partial succession from which drawing 10 differs. In this example, the "Hybrid Vehicle" class 110 is displayed like the child class of the "Sedan" class 111. Partial succession of the "HybridVehicle" class 110 which is a low-ranking class is carried out by not a regular child class but cay SUOBU (CaseOf) of the "Sedan" class 111 rather than the "Sedan" class 111. In order to show this clearly, the reference (Ref) mark 112 is given. The layered structure indicator 1 is provided with the mechanism jumped to a regular class when a class with this Ref mark is chosen, and the user can search also considering the "Sedan" class 111 as a starting point.

[0049]

"A display according to color"

Drawing 12 is a figure for explaining the display according to color in a layered structure display. The color of each hierarchy of the layered structure displayed by the layered structure indicator 1 and the color of the top mark of the property list in the property list indicator 3 are displayed in the same color. According to such a display according to color, it can be checked whether each property in a property list is defined by which hierarchy of the higher rank class seen from the present selected class 121 at a glance, and a very good visual effect is acquired.

[0050]

"Partial succession former class reference display"

Drawing 13 is a figure showing a reference display in the source class [class / of a partial succession place] of the property which carried out partial succession. As for the reference number 21, in drawing 13, a property list display and 23 are the examples of a property information display a layered structure display and 22. The "Hybrid Vehicle" class in the layered structure display 21 and the "engine_type" property in the property list display 22 are items chosen by the user, and highlighting is carried out.

[0051]

The C mark 25 shown in the left end of the property currently displayed in the property list display 22 is a mark for specifying that it is what requires this property for partial succession. A user is that this "Hybrid Vehicle" class refers to the property information display 23 for the detailed information of the "engine_type" property which is carrying out partial succession, and it can check that this property is inherited from a "Sedan" class. If a user pushes the Jump (jump) button 24, a focus will change to the source class "Sedan" class of the property which carried out partial succession in the layered structure display 21 here, Renewal of a display of the property list in the property list display 22 is carried out simultaneously with this at the property list of a "Sedan" class.

[0052]

"Virtual (imagination) and route table Shimesu"

The layered structure display 21 usually displays one layered structure. This is equivalent to the tree which makes a source route (ROOT) without a higher rank class. Two or more tree views are performed by the change of a screen in the graphic library established in C++ known well, a Java language, etc. For this reason, only one tree of them can be displayed at once. On the other hand, in the embodiment of this invention, the route of each tree is expressed as a low rank class of the created virtual route class by providing a virtual route to two or more trees, and creating the class corresponding to it.

[0053]

Drawing 14 shows the display example at the time of providing the virtual route in a layered structure indicator. The virtual route 140 is treated on a par with the usual class. Under the virtual route 140, all the functions concerning this embodiment described so far can usually be used like a class.

[0054]

For example, drawing 15 is set up display only the route 151,152 of each tree under the ground-floor layer of the virtual route 150 the first stage.

[0055]

Drawing 16 is a figure showing the example of setting out of an initial deployment class. Here, the default deployment number of stories is set to "TREE_OPEN=6." Each class which indicates by deployment at the time of the first stage can be specified below as "TREE_OPEN_CLASS" by enumerating the identifiers 160 of a class. Although carried out a principle [a default deployment number of stories], if deployment of a class has specification of a class in this way, it will be considered as deployment to this class about the branch parts.

[0056]

According to the embodiment of this invention explained above, it arranges to an indicator so that the information about the information about the class containing existence, a property list, etc. of not only the information (inclusion relation) about the layered structure itself but an instance and a property may be shown effectively. Therefore, the interface which a user tends to operate can be provided. For example, a user or a donor can specify intentionally the class range which carries out a hierarchical display more, it can also be limited, a complicated layered structure can be extracted only to a required portion, and it can provide for a user. The search which does not require load for a system can also be made to perform by limiting search start points and providing for a user.

[0057]

This invention is not limited to the embodiment mentioned above, but changes variously, and is feasible. For example, this invention is applicable not only to a hierarchical database which succession generates between classes but the display of the layered structure in various systems. For example, this invention is applicable also to the display of the directory structure tree in a file system. For example, it is specifying the existence of the instance for every directory etc.

[0058]

[Translation done.]

* NOTICES *

JPO and INPIT are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

TECHNICAL FIELD

[Field of the Invention]

This invention relates to the suitable hierarchical structure display device and method for a hierarchical database.

[0002]

[Translation done.]

description of drawings(2004-178015)

* NOTICES *

JPO and INPIT are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]

[Drawing 1] The block diagram showing the outline composition of the layered structure display system concerning one embodiment of this invention

[Drawing 2] The figure showing the inclusion relation between the classes concerning the above-mentioned embodiment

[Drawing 3] The flow chart which shows an example of the procedure of an existence judging of the instance in the low rank class concerning the above-mentioned embodiment

[Drawing 4] The figure showing the example of composition of the more concrete GUI picture about the layered structure display concerning the above-mentioned embodiment

[Drawing 5] The figure showing the display example of the detailed information by the property information indicator concerning the above-mentioned embodiment

[Drawing 6] The figure showing an example of the heritage (Heritage) display concerning the above-mentioned embodiment

[Drawing 7] The figure for explaining the traversal (Traversal) search concerning the above-mentioned embodiment

[Drawing 8] The flow chart which shows an example of procedure which sets up automatically the search start points concerning the above-mentioned embodiment

[Drawing 9] The figure showing an example of the layered structure accompanied by the multi-inheritance concerning the above-mentioned embodiment

[Drawing 10] The figure showing the display example of the partial succession in the layered structure concerning the above-mentioned embodiment

[Drawing 11] The figure showing other display examples of the partial succession concerning the above-mentioned embodiment

[Drawing 12] The figure for explaining the display according to color concerning the above-mentioned embodiment

[Drawing 13] The figure showing a reference display in the source class [class / of the partial succession place concerning the above-mentioned embodiment] of the property which carried out partial succession

[Drawing 14] The figure showing the display example at the time of providing the virtual route in the layered structure indicator concerning the above-mentioned embodiment

[Drawing 15] The figure showing the initial display example at the time of the virtual route setting concerning the above-mentioned embodiment

[Drawing 16] The figure showing the example of setting out of the initial deployment class concerning the above-mentioned embodiment

[Description of Notations]

- 1 -- Layered structure indicator
- 2 -- Class information indicator
- 3 -- Property list indicator
- 4 -- Property information indicator
- 5 -- CPU
- 6 -- Memory
- 7 -- Secondary memory
- 8 -- Display

description of drawings(2004-178015)

[Translation done.]

* NOTICES *

JPO and INPIT are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

EFFECT OF THE INVENTION

[Effect of the Invention]

As explained above, according to this invention, the hierarchical structure display device and method of displaying appropriately and effectively the inclusion relation between the classes in a hierarchical database can be provided.

[Translation done.]

* NOTICES *

JPO and INPIT are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

PRIOR ART

[Description of the Prior Art]

the technical field of today and computer software -- more -- a user -- progress is accomplished every day that a FRIENDLY function should be provided. For example, so that operation (operation) in which a user moves to a desired directory in use of a file system, and a desired file is opened can be performed visually and intuitively. The graphic user interface (GUI) can display directory (folder) structure on tree form combining a line, an icon, etc., and can show it to a user. In general-purpose operating systems, such as the operating systems (os) (trademark) Windows (trademark), UNIX (trademark), and LINUX of Microsoft Corp., such a GUI function is adopted widely.

[0003]

About between each node (directory) of a directory structure tree, there are no relations between the information, including file etc., included in the node of a higher rank and the information included in a low-ranking node, such as inclusion or a subset. That is, each node which forms a directory structure tree and which begins from a root node only expresses the interconnection relation of the container which dedicates information, including a file etc.

[0004]

On the other hand, the database which makes representation an object oriented database (OODB) and an Object Relational Database (ORDB) has a layered structure, and has a mechanism in which a subdivided classification inherits the attribute of an upper classification in this layered structure. In such a database, there is the feature that an attribute accumulates by low-ranking classification by succession. It is also called "inheritance" that a subdivided classification inherits the attribute of an upper classification, and such art is indicated in much literature (for example, refer to the following nonpatent literature 1.).

[0005]

In an object oriented database (OODB)-related technical field, the classification in a hierarchy is called a "class" in many cases. In this specification, a "classification" and a "class" are used as a term which has the almost same meaning. In an object relational database (ORDB), the table which allowed succession is equivalent to a class. Between the tables in the hierarchical order, an attribute is inherited from the table of a higher rank to a low-ranking table. An attribute here is equivalent to the header information of the column which constitutes a higher rank table in ORDB, and this is inherited to a low order table.

[0006]

In this specification, a "hierarchical database" is called including both an object oriented database (OODB) and an Object Relational Database (ORDB). Data with the same attribute kind belonging to each hierarchy's class is called an "instance", and the set is called "population" of data.

[0007]

Thus, an above-mentioned directory structure tree differs in the character clearly from the layered structure of a hierarchical database having the inclusion relation (succession relation) between classes.

[0008]

The service which applied the hierarchical database to the product catalog data base is already provided partly (see the following nonpatent literatures 2 and 3.).

[0009]

prior art(2004-178015)

[Nonpatent literature 1]
Object-Oriented Concepts, Databases, and Applications, Edited by Won Kim, 1989, ACM
Press
[0010]
[Nonpatent literature 2]
An incorporated company portal, the Internet <URL:<http://www.portalcorp.net>>
[0011]
[Nonpatent literature 3]
e-ingBiz.com, the Internet <URL:[http://www.e-ingbiz.com//CATALOG2 /](http://www.e-ingbiz.com//CATALOG2/)
[servlet/CatalogSearch?Lang=ja](http://www.e-ingbiz.com//CATALOG2 /servlet/CatalogSearch?Lang=ja)>
[0012]

[Translation done.]

* NOTICES *

JPO and INPIT are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

TECHNICAL PROBLEM

[Problem(s) to be Solved by the Invention]

with the Explorer of a display [in / about a layered structure display / the conventional directory structure tree] Corp., for example, Microsoft, (trademark)

- Indicate the low-ranking folder (directory) by deployment according to the opening-and-closing demand to the folder of a higher rank.
- By stage lowering, the folder twist expresses that it is a low-ranking folder to the folder of a higher rank.
- Make it by arranging stage lowering with every [of a hierarchy] level (depth) show easily that a level is the same folder.

What was said can be performed.

[0013]

However, these enable it to follow the connecting relation of a directory (node) to the last.

[0014]

Therefore, such a conventional layered structure display is not suitable for the hierarchical database that a property (attribute) is inherited from the class of a higher rank to a low-ranking class.

[0015]

The layered structure display in a hierarchical database is urged to be able to express the inclusion relation between classes in case each class expresses the concept characterized with the property to own. When the layered structure of a hierarchical database is constituted by the class, it is preferred that the existence of the instance of this class can be displayed.

[0016]

This invention is made in consideration of this situation, and is a thing. The purpose is to provide the hierarchical structure display device and method of displaying the inclusion relation between the classes which can be boiled and set appropriately and effectively.

[0017]

[Translation done.]